Lab 10: Complete APIs Integration (Add, Edit, and Delete)

You will continue to **update the front-end** of the web-based FilmLibrary to **invoke all the APIs** designed and implemented in Lab 8. Remember to minimize the "*n-clients problem*" shown during the lecture, by reloading the application state every time you modify the FilmLibrary on the server.

1. Add a New Film

When the user adds a new film, it must be **saved on the server-side database**, and the displayed list of films must be updated accordingly. To do so, invoke the proper API endpoint for adding films, and then retrieve the updated list of films from the server. In this way you will be sure that client and server side are always aligned. Execute every add operation as performed by the user with id 1.

2. Edit a Film

Make the update operations persistent: when the user updates a film through its dedicated edit form, the **film is modified on the server-side database**, and the list of films is updated and displayed accordingly. To do so, invoke the proper API for updating a film, and then retrieve the updated list of films from the server.

3. Optional: Edit in-line and Delete Films

- Users should have the possibility of changing some values "in-line":
  - the *favorite* property through the checkbox displayed near each film in the list.
  - the rating property through the displayed stars. E.g., if the film is rated two out of five and the user clicks on the fourth star, the new rating becomes four out of five. **Note**: if the user clicks on the current assigned rate, you should not trigger an update in the back-end.
- Make the delete persistent: when the user deletes a film, the **film is removed from the server-side database**, and the list of films is updated and displayed accordingly. To do so, invoke the proper API for deleting a film, and then retrieve the updated list of films from the server.

Hints:

1. ***Trust the server!*** *It shall be always up-to-date, and every operation rely on it, not on the internal state of the web application.*
2. *Create a back-up copy of the database before testing your APIs.*